

(19) 日本国特許庁 (J P)

(12) 特 許 公 報 (B 2)

(11) 特許番号

第2862388号

(45) 発行日 平成11年(1999) 3月3日

(24) 登録日 平成10年(1998)12月11日

(51) Int.Cl.<sup>8</sup> 識別記号  
 G 0 6 T 5/20  
 G 0 6 F 15/16 3 9 0  
 G 0 6 T 1/20

F I  
 G 0 6 F 15/68 4 0 0 J  
 15/16 3 9 0 Z  
 15/66 K

請求項の数1(全 8 頁)

(21) 出願番号 特願平3-57118  
 (22) 出願日 平成3年(1991) 3月20日  
 (65) 公開番号 特開平4-291682  
 (43) 公開日 平成4年(1992)10月15日  
 審査請求日 平成10年(1998) 2月18日

(73) 特許権者 396020800  
 科学技術振興事業団  
 埼玉県川口市本町4丁目1番8号  
 (72) 発明者 今井正治  
 愛知県豊橋市三ノ輪町3丁目89番地の2  
 サーラシティ東向山603  
 (72) 発明者 富田穠太  
 愛知県豊橋市橋良町東郷20-4 ハイッ  
 サーティワン3-A号  
 (74) 代理人 弁理士 蛭川 昌信  
 審査官 新井 則和

最終頁に続く

(54) 【発明の名称】 超高速画像処理システムのフィルタリング処理方式

1

(57) 【特許請求の範囲】

【請求項1】 1行分の画像データの全部または一部をラスタースキャン順に取り込む複数の入力要素からなる入力ユニットと、各入力要素からの画像データが同時に転送され、画素単位で並列的に画像処理演算を行う複数の処理要素からなる処理ユニットと、各処理要素からの処理データが同時に転送される複数の出力要素からなる出力ユニットと、入力ユニット、処理ユニット、出力ユニットを制御するコントローラとを備え、順次各行毎に画素単位で並列的に左右の対称性を持つ重みを有するフィルタにより画像処理を行う超高速画像処理システムであって、各処理要素は処理を受け持つ列の各画素データにフィルタの中心の列の各重みを乗算し、乗算結果を互いに加算して保持し、さらに処理を受け持つ列の各画素データにフィルタの中心の列の片側の他の列の各重みを

2

乗算し、乗算結果を互いに加算して保持し、各処理要素は他の列の処理要素が保持しているデータを転送により受け取ってフィルタリング処理を実行することを特徴とする超高速画像処理システムのフィルタリング処理方式。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は大規模画像の前処理を実時間で行うことができる超高速画像処理システムR I P E (Real-time Image Processing Engine) のフィルタリング処理方式に関するものである。

【0002】

【従来の技術】 医療、工業生産などの分野で計算機を用いた画像処理の必要性が高まって来ている。これらの応

## 3

用分野における近い将来の目標は2次元またはそれ以上の多次元大規模画像（高画質画像）の実時間処理である。システムの認識能力を向上させるためには画像自体の分解能をあげることが必須であるが、そのためには、画素数を増大させること、および前処理でのフィルタのマスクサイズを増大させることが必要である。

【0003】ここ数年間での画像処理に対する具体的要求の例として以下の3項目が考えられる。

【0004】1画像当たり2048×2048以上の画素を持つ多値およびカラーの画像の処理が可能となること。

【0005】上記の画像に対してフィルタリングなどの局所並列処理が高速に行えること、および50×50程度の大きさのマスクを用いたフィルタリング処理が可能であること。

【0006】画像処理ではフィルタリングを中心とする前処理が行われる場合が多く、画像の規模およびフィルタの規模が増大するのに従い、前処理に必要な計算時間は急激に増加する。

【0007】これまで医療、工業生産などで実用化されている画像処理システムでは、512×512程度の画素を持つ画像を対象としているが、2048×2048画素程度の解像度を持つ医療用X線フィルムなどの複雑な濃淡画像の前処理を行うためには、スーパーコンピュータなどの汎用大型計算機を用いても実時間処理は困難である。このような大規模画像の前処理を効率よく行う専用システムを実現するためには、処理の並列化およびパイプライン化が有効であると考えられる。特にフィルタリング処理のアルゴリズムの多くは並列型の積和演算を頻繁に用いているので、画素単位での空間並列処理が効果的であると考えられる。

【0008】

【発明が解決しようとする課題】ところで、画像1行中の画素数と同じ個数の処理要素を用意して画素単位での並列処理を行う、所謂行並列型局所演算におけるフィルタリング処理は、各処理要素において自分が処理を受け持つ列の画素データについての演算を行うとともに、他の列の画素データを転送してもらい、そのデータにフィルタの重みを乗算し、加算する等の処理を行うために各列間の転送回数、乗算回数、加算回数が多くなり、特にフィルタのマスクサイズが大きくなった場合にはフィルタリング処理に時間を要してしまうという問題があった。

【0009】本発明は上記課題を解決するためのもので、行並列型局所演算における乗算回数、加算回数、列間のデータ転送回数を大幅に低減化してフィルタリング処理時間を短縮することができる超高速画像処理システムのフィルタリング処理方式を提供することを目的とする。

【0010】

## 4

【課題を解決するための手段】本発明は、1行分の画像データの全部または一部をラスタースキャン順に取り込む複数の入力要素からなる入力ユニットと、各入力要素からの画像データが同時に転送され、画素単位で並列的に画像処理演算を行う複数の処理要素からなる処理ユニットと、各処理要素からの処理データが同時に転送される複数の出力要素からなる出力ユニットと、入力ユニット、処理ユニット、出力ユニットを制御するコントローラとを備え、順次各行毎に画素単位で局所並列的に左右の対称性を持つ重みを有するフィルタにより画像処理を行う超高速画像処理システムであって、各処理要素は処理を受け持つ列の各画素データにフィルタの中心の列の各重みを乗算し、乗算結果を互いに加算して保持し、さらに処理を受け持つ列の各画素データにフィルタの中心の列の片側の他の列の各重みを乗算し、乗算結果を互いに加算して保持し、各処理要素は他の列の処理要素が保持しているデータを転送により受け取ってフィルタリング処理を実行することを特徴とする。

【0011】

【作用】本発明は画像1行中の画素数と同じかそれ以下の個数の処理要素を用意して画素単位での並列処理を行う行並列型局所演算により、左右の対称性を持つフィルタを使用する画像処理システムにおいて、各処理要素は処理を受け持つ列の画素データにフィルタの中心の列の各重みを乗算して加算するとともに、処理を受け持つ列の画素データにフィルタの中心の列より片方の側の他の列の各重みを乗算して加算し、各処理要素は他の列の処理要素が保持している演算結果を転送により受け取ってフィルタリング処理を実行することにより、フィルタリング処理における乗算回数、加算回数、転送回数を大幅に低減化して超高速での処理を行うことが可能となる。

【0012】

【実施例】図1は本発明の演算処理を説明するための図、図2は本発明のハードウェア構成を示す図、図3は各画像処理要素を示す図である。図中、4はスリットメモリ、10は入力ユニット(LIU)、10-1~10-nはラッチ回路(LE)、20は処理ユニット(LPU)、20-1~20-nは処理要素(PE)、30は出力ユニット(LOU)、30-1~30-nはラッチ回路(OE)、40はホストコンピュータ、50は外部コントローラ、21-iはセレクタ(Selector)、22-iは算術論理ユニット(ALU)、23-iはレジスタファイル、24-iはフラグレジスタ、25-iは通信コントローラ、26-iはバスである。

【0013】まず、本発明の超高速画像処理システムRIPEを図2、図3により説明する。

【0014】本発明のRIPEでは、65536階調(16ビット)の濃淡画像データに対する各行の処理を、入力-演算-出力の3つのステージに分割し、パイプライン的に処理することにより画像データの入出力と

5

演算処理を平行して行うものであり、演算ステージでは画像1行中の画素数と同じ個数のPE (Processing Element) を用い、外部コントローラより各PEに対して同一の命令を1つずつ与え、それぞれの画像データに対して同一の処理を行うSIMD (Single Instruction stream Multiple Data stream) 型の並列処理が行われるため、画像データの処理が1行分同時に行われる。

【0015】本発明のRIPEのハードウェア構成は図2に示すようなものであり、このシステムはホストシステムのバックエンドプロセッサとして動作し、外部コントローラ50、入力ユニット10、演算ユニット20、出力ユニット30からなっている。外部コントローラ50はホストコンピュータ40との同期をとりながら、入力ユニット10、演算ユニット20、出力ユニット30の制御を行い、ユーザーが作成した処理プログラムを格納するRAMと、予め基本的な処理のプログラムが格納されているROMを持ち、ホストコンピュータ40からの指示にしたがってRAMまたはROMに記憶された命令を1ステップずつ順次各演算ユニット20に送っており、各演算ユニットは命令されたことだけを実行する処理機械として機能する。なお、ユーザーが作成した処理プログラムは処理に先立ち、あらかじめホストコンピュータ40からコントローラ50のRAMにダウンロードされる。

【0016】入力ステージを受け持つLIU10は16ビット幅のn個のラッチ回路から構成されてシフトレジスタとして動作し、他のメモリに記憶されているイメージデータ、或いはカメラで読み込んだイメージデータがラスタースキャン順に入力され、その画素データを順次シフトし、画像1行分の画素データが揃った時点で1行分の画素データを同時にLPU20の各PE20-1~20-nに同時並列的に転送する。

【0017】LPU20はn個のPEから構成されて演算ステージを受け持つっており、図3に示すような各モジュールからなっている。図3はi番目のPE20-iを示したものであり、ラッチ回路10-iからのデータを一部切り出して記憶装置4 (本発明ではスリットメモリと称する) に読み込み、このデータをバス26-iを通してALU22-iで演算して中間結果をレジスタ23-iに格納し、また結果をセクタ21-iを通してラッチ回路30-iへ出力するものである。各PEはコントローラ50からの命令を1ステップずつ受け取って一斉に同一処理を行っており、自身のメモリにはプログラムが格納されておらず、外部からの指令によって単に処理機械として動作する。

【0018】スリットメモリ4は、フィルタリング処理に必要なデータを格納するためのものである。すなわち、画像データに対する局所並列処理では1つの画素の

6

出力値を決定するために、その画素の近傍の画素データも必要となり、この場合全てのPEが各自必要なデータを内部に持つこととすると、システム全体ではデータが重複し不経済である。そこで、各PE内でのスリットメモリにそのPEが処理を受け持っている列のデータを必要な個数分、すなわちマスクの縦の画素の個数分だけ記憶させることにし、残りの近傍データは他のPE内のスリットメモリに保持されているので、隣接するPE間で通信コントローラ25-iを通してデータ転送を行うことによって得るようにする。本実施例では各スリットメモリは16ビット幅の64個のセルからなり、列方向の長さが64以下のマスクを用いた局所並列処理が実現できる。

【0019】ALU22-iは各画素に対して16ビット幅の算術論理演算処理を行うものである。処理内容はどのようなフィルタ処理を行うかにより異なるが、ALU22-iは外部コントローラから1つずつ与えられる命令を実行する。

【0020】レジスタファイル23-iは中間結果等のデータを格納するレジスタが割付けられたファイルで、16個の16ビット幅のGR (General Register)、通信コントローラ25-iを通して隣接するPE間で転送されるデータを格納する通信用レジスタCR (Communication Register) 等からなり、またフラグレジスタ24-iは符合、零、オーバーフロー、キャリーなどのフラグの内容を格納するためのものである。

【0021】セクタ21-iは局所並列処理を行う場合、画像の外周部では近傍のデータが完全には得られないため計算結果は無効となり、従来の画像処理アルゴリズムでは、通常強制的に出力値を0にしたり、処理内容に応じて適切な定数または近傍の値等を設定するようにしているが、この画像外周部の出力値を定数にするか、無効ではあるが計算値にするかいずれかを設定できるようにし、この機能を行っている。

【0022】こうして各PEはスリットメモリ4に読み込まれた必要なデータを順次読み出すと共に、隣接する処理ユニットからのデータを通信コントローラを通して受け取り、フィルタリング処理を行いラッチ回路に出力している。

【0023】図2のLOU30は16ビット幅のn個のラッチ回路からなり、出力ステージを受持ち、LPU20で演算されたデータは1行分同時にLOU30に転送され、その後順次シフトされることによって1画素づつラスタースキャン順に出力される。

【0024】このようなRIPEシステムにおいて、図1(b)に示すような左右対称の3×3のフィルタによる処理を行う場合を考える。図1(a)は中心が(i, j)である3×3の画素データであり、行われる演算は図1(a)と図1(b)のマトリックスの要素ごとの積

10

20

30

40

50

7

で得られた次のマトリックスの各要素の和であるとする。

【0025】

$$\begin{bmatrix} a x_{i-1j-1} & d x_{ij-1} & a x_{i+1j-1} \\ b x_{i-1j} & e x_{ij} & b x_{i+1j} \\ c x_{i-1j+1} & f x_{ij+1} & c x_{i+1j+1} \end{bmatrix}$$

この演算を行う  $i$  列の処理要素を  $PE_i$  とすると、 $PE_i$  はスリットメモリより画素データを読み出して演算を行い、結果をレジスタに格納する。この演算は次のようになる。

【0026】  $i$  列の各重みと画素データとの乗算処理

( $d x_{ij-1}$ ,  $e x_{ij}$ ,  $f x_{ij+1}$ )

加算処理 ( $d x_{ij-1} + e x_{ij} + f x_{ij+1}$ )

( $i-1$ ) 列の画素データの転送処理 ( $x_{i-1j-1}$ ,  $x_{i-1j}$ ,  $x_{i-1j+1}$ )

( $i-1$ ) 列の画素データと重みの乗算及び乗算結果の加算処理 ( $a x_{i-1j-1} + b x_{i-1j} + c x_{i-1j+1}$ )

加算処理 (+)

( $i+1$ ) 列の画素データの転送処理 ( $x_{i+1j-1}$ ,  $x_{i+1j}$ ,  $x_{i+1j+1}$ )

( $i+1$ ) 列の画素データと重みの乗算及び乗算結果の加算処理 ( $a x_{i+1j-1} + b x_{i+1j} + c x_{i+1j+1}$ )

加算処理 (+)

この演算における乗算回数は  $3 \times 3 = 9$  回、加算回数は 8 回、転送回数は 6 回であり、 $M \times M$  のマスクサイズの場合には、それぞれ乗算回数  $M^2$  回、加算回数  $M^2 - 1$  回、転送回数  $M(M-1)$  回となり、 $M$  が大きくなると多くの処理時間を要してしまうことになる。

【0027】そこで、本発明ではフィルタの重みが左右対称であることを利用し、各  $PE_i$  での処理負担を軽減化することを考える。

【0028】図 1 (c) に示すように処理要素  $PE_{i-1}$  は、自身が処理を受け持つ列の画素データをスリットメモリより読み出し、重み  $d$ ,  $e$ ,  $f$  をそれぞれ乗算して互いに加算するとともに、同じ列の画素データに対して重み  $a$ ,  $b$ ,  $c$  を乗算して同様に加算し、

$d x_{i-1j-1} + e x_{i-1j} + f x_{i-1j+1}$

$a x_{i-1j-1} + b x_{i-1j} + c x_{i-1j+1}$

の計算結果をそれぞれレジスタに格納する。

【0029】処理要素  $PE_i$  は処理を受け持つ列の画素データをスリットメモリより読み出し、重み  $d$ ,  $e$ ,  $f$  を乗算して加算するとともに、同じ列の画素データに対して重み  $a$ ,  $b$ ,  $c$  を乗算して加算し、

$d x_{ij-1} + e x_{ij} + f x_{ij+1}$

$a x_{ij-1} + b x_{ij} + c x_{ij+1}$

の計算結果をレジスタに格納する。

【0030】処理要素  $PE_{i+1}$  は処理を受け持つ列の画素データをスリットメモリより読み出し、重み  $d$ ,  $e$ ,  $f$  を乗算して加算するとともに、同じ列の画素デー

8

タに対して重み  $a$ ,  $b$ ,  $c$  を乗算して加算し、

$d x_{i+1j-1} + e x_{i+1j} + f x_{i+1j+1}$

$a x_{i+1j-1} + b x_{i+1j} + c x_{i+1j+1}$

の計算結果をレジスタに格納する。

【0031】このように、各列の処理要素は自身が受け持つ列の各画素データに対してフィルタの中心の列の各重みを乗ずるとともに、フィルタの中心の列の片方の側の列の重みも乗算して各々加算する。そして、各処理要素での演算においては、隣接する列から、その列の画素データに対して重み  $a$ ,  $b$ ,  $c$  を乗算した結果のデータを転送してもらって演算処理する。例えば、図 1 (c) の処理要素  $PE_i$  では図の矢印で示すように隣接する  $PE_{i-1}$  から

$a x_{i-1j-1} + b x_{i-1j} + c x_{i-1j+1}$

の計算結果を受け取り、また隣接する  $PE_{i+1}$  から

$a x_{ij-1} + b x_{ij} + c x_{ij+1}$

の計算結果を受け取ることにより、フィルタリング処理の演算を行うことができる。この結果、処理要素  $PE_i$  で行う乗算回数は 6 回、加算回数は 6 回、転送回数は 2 回と低減化することができる。この方式をより大きいマスクサイズに適用する場合には、フィルタの中心の列でない列の数が増えるだけで、これらの列の各重みを処理を受け持つ列のデータに乗算すればよい。この方式を  $M \times M$  のマスクサイズに適用した場合には、それぞれ乗算回数は  $M(M+1)/2$  回、加算回数は  $(M-1)(M+3)/2$  回、転送回数は  $(M^2 - 1)/4$  回となり、処理時間を短縮することができる。

【0032】次に、左右対称の特殊な例であるラプラシアンフィルタの場合、例えば、

-1	2	-1
2	-4	2
-1	2	-1

のような重みを持つラプラシアンフィルタの場合には、右辺の重みの  $-1$ ,  $2$ ,  $-1$  を  $-2$  倍すれば中心の重みを求めることができるので、各処理要素  $PE$  はそれ自身が処理を受け持つ列の各画素データに対しては  $-1$ ,  $2$ ,  $-1$  の重みを乗算して加算し、自分自身は得られた和を  $-2$  倍して用い、隣の列へ加算の結果をそのまま渡せばよいことになり、乗算回数、加算回数をさらに減らすことができ、 $M \times M$  のマスクサイズの場合には  $M^2$  の乗算回数を  $(M+1)$  回に、 $M^2 - 1$  の加算回数を  $2(M-1)$  回に、 $M(M-1)$  回の転送回数を  $4M/3 - 2$  回にそれぞれ低減化することが可能である。

【0033】さらに、図 1 (b) のマスクの重みが全て同じであり、例えば

30

40

50

a	a	a
a	a	a
a	a	a

のように一様な場合には、列方向は前述した方法を適用し、行方向については、1行ずらしたとき2行分は前回のデータがそのまま使用できるので、これによりさらに演算の省略ができ、 $M \times M$ のマスクサイズの場合では、

$(M^2 - 1)$  回の加算回数を  $2(M - 1)$  回に、 $M(M - 1)$  回の転送回数を  $(M - 1)$  回にそれぞれ低減化することができ、転送回数  $M(M - 1)$  回となる。

【0034】図4は局所平均化、疑似メディアン、局所最大値フィルタについて本発明を適用した場合のマスクサイズとフィルタリング処理時間との関係を示したものである。

【0035】特性A、B、Cは従来法によるもの、D、E、Fは本発明によるものであり、例えば局所最大値フィルタの場合にはマスクサイズ  $63 \times 63$  の場合、従来法における内部処理時間が  $837.35 \text{ ms}$  であるのに対して、本発明における処理時間は  $7.68 \text{ ms}$  となり、約1%にまで短縮され、超高速処理に好適であることがわかる。

【0036】なお、本発明の画像処理は、図5に示すように、カメラ63で直接読み込んだ画像データをA/D変換して直接RIPEシステム60に読み込むか、あるいは一旦メモリ61に読み込んだデータを読み出してRIPEシステムに読み込むか、どちらの処理を行っても良く、この結果をモニタ64に出力し、あるいは結果を再度メモリ61に書き込む等ホストコンピュータ40からの指示により実行することができる。

【0037】また、上記実施例ではPEが画像1行分の画素分だけ用意されて1行分の画像処理が同時並列的に

実行される場合について説明したが、本発明はこれに限定されるものではなく、PEが画像1行分の画素に満たない場合でも、順次PEを一部重ねながらずらしていくことによって対応可能であり、この場合にフィルタのマスクサイズに応じて重なり度合を適宜変更すればよい。

【0038】

【発明の効果】以上のように本発明によれば、各列に対応する処理要素は自身が受け持つ列の画素データに対してフィルタの中心の列の各重みを乗算して互に加算する演算を行うとともに、自身が受け持つ列の画素データに対してフィルタの中心の列の片方の側の列の各重みを乗算して互に加算する演算を行い、演算結果を各列の処理要素で互いに転送して利用することにより、乗算回数、加算回数、転送回数を低減化してフィルタリング処理時間を大幅に短縮することが可能となる。

【図面の簡単な説明】

【図1】 本発明におけるフィルタリング処理を説明するための図である。

【図2】 本発明におけるメモリアクセスのハードウェア構成を説明するための図である。

【図3】 各画像処理要素を示す図である。

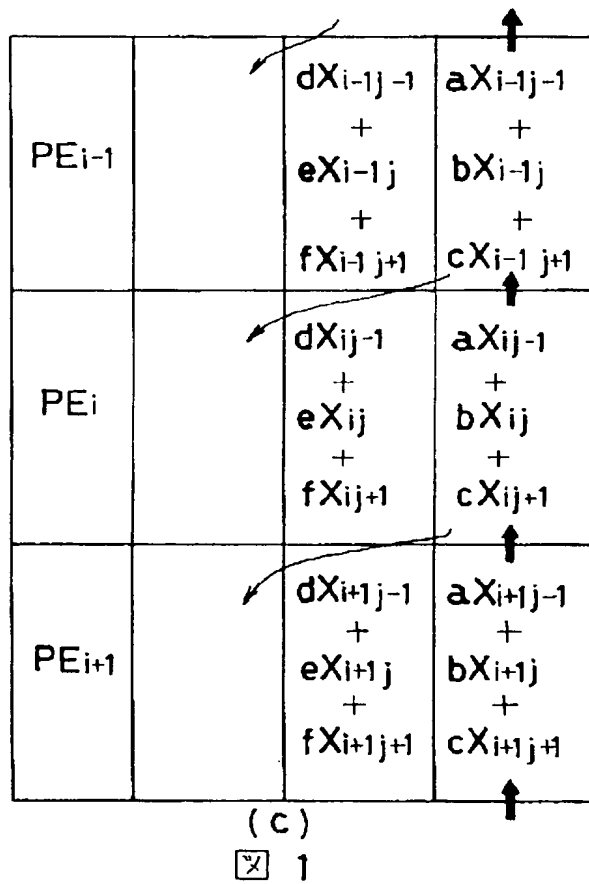
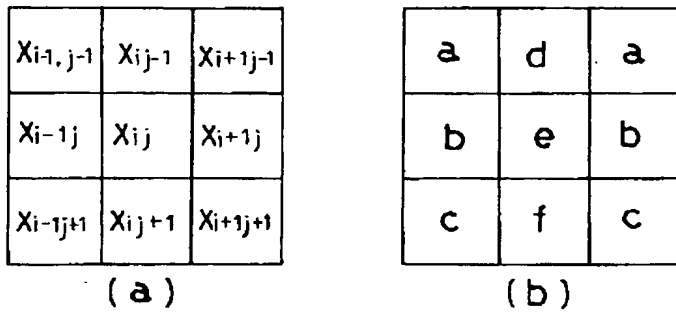
【図4】 マスクサイズと実行時間との関係を説明する図である。

【図5】 RIPEシステムを説明するための図である。

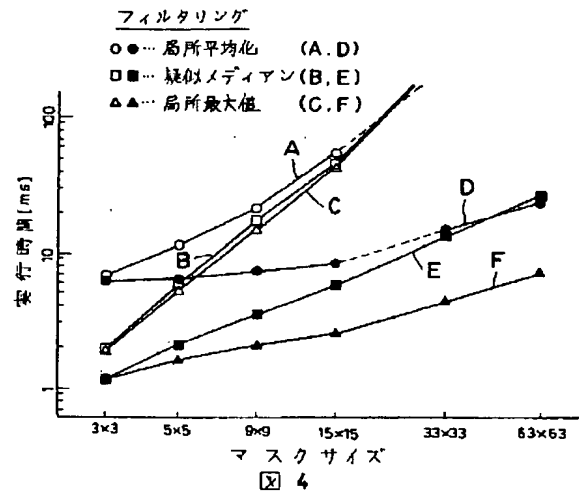
【符号の説明】

4…スリットメモリ、10…入力ユニット(LIU)、  
10-1~10-n…ラッチ回路、20…処理ユニット(LPU)、  
20-1~20-n…処理要素(PE)、  
30…出力ユニット(LOU)、30-1~30-n…ラッチ回路、  
40…ホストコンピュータ、50…外部コントローラ、  
21-i…セレクト(Selector)、22-i…算術論理ユニット(ALU)、  
23-i…レジスタファイル、24-i…フラグレジスタ、  
25-i…通信コントローラ、26-i…バス。

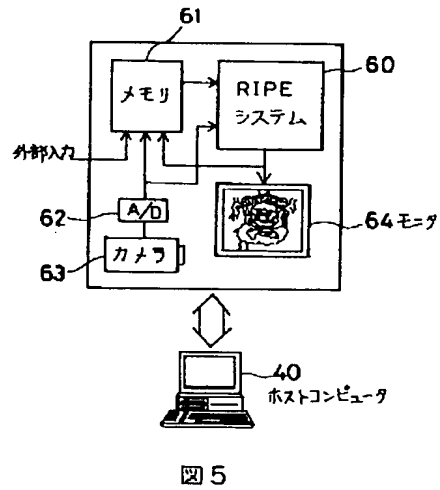
【図1】



【図4】



【図5】



【図2】

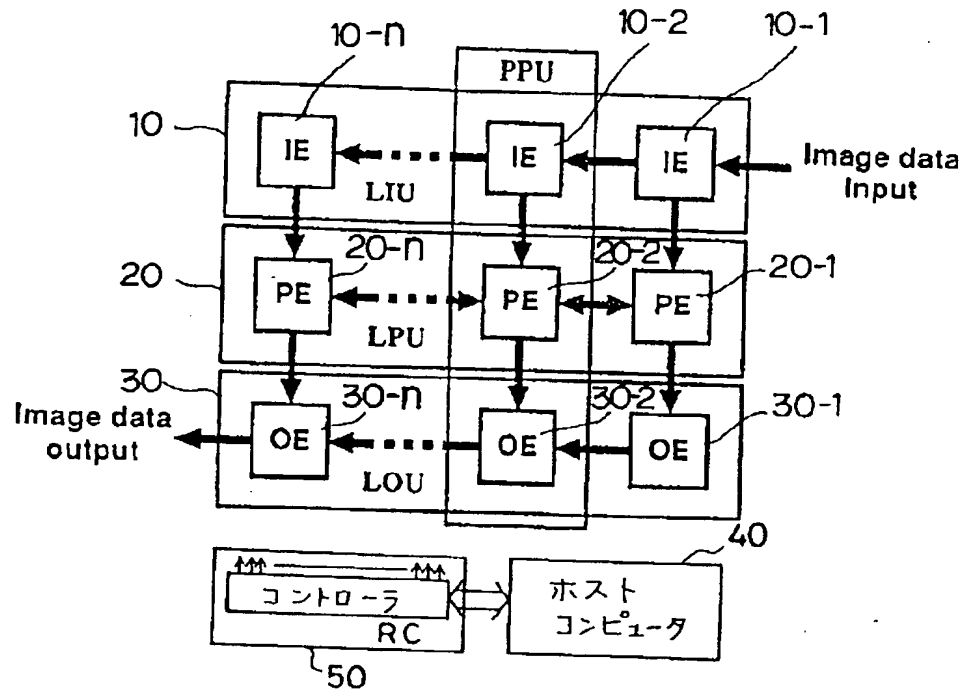
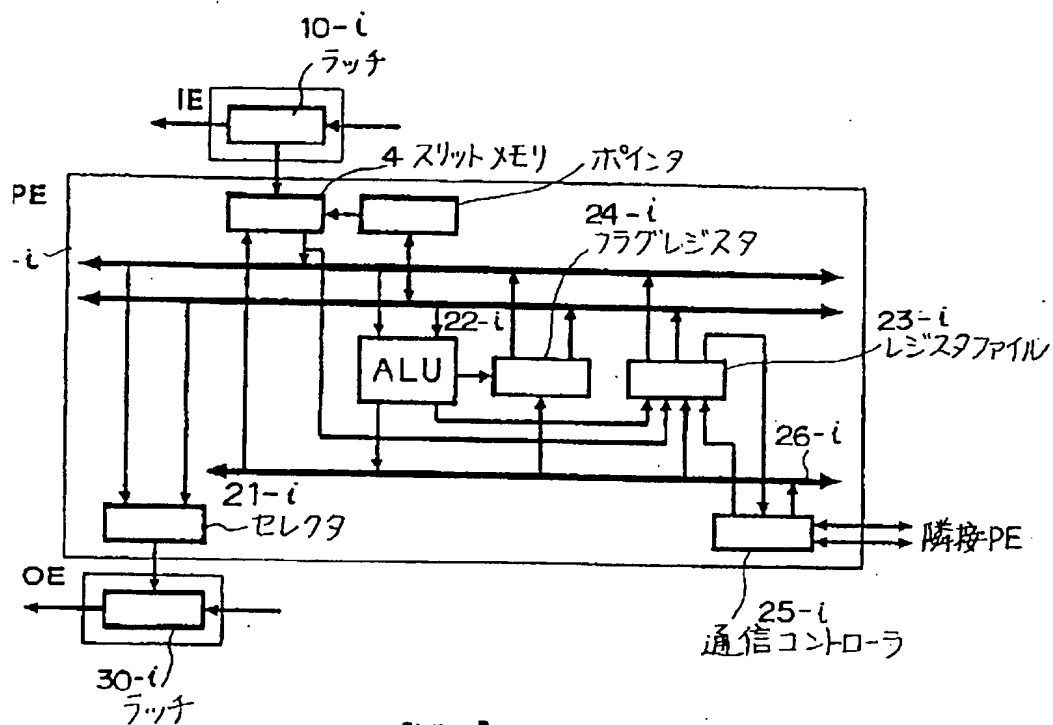


図 2

【図3】



【図3】

フロントページの続き

(56) 参考文献 特開 昭63-118885 (J P, A)  
 特開 昭58-217072 (J P, A)  
 特開 平2-25987 (J P, A)  
 特開 平2-300981 (J P, A)  
 特開 昭63-24469 (J P, A)

(58) 調査した分野(Int. Cl.<sup>6</sup>, D B 名)  
 G06T 5/20  
 G06F 15/16 390  
 G06T 1/20